

Welcome to IEEE Xplore®

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

 Print Format

Your search matched **13** of **1041798** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance in Descending** order.

Refine This Search:

You may refine your search by editing the current search expression or entering a new one in the text box.

depth first traversal

 Check to search within this result set

Results Key:

JNL = Journal or Magazine **CNF** = Conference **STD** = Standard**1 Implicit depth-first state traversal of sequential machines***Ghosh, A.; Devadas, S.;*

Circuits and Systems, 1991., IEEE International Symposium on , 11-14 June 1991

Pages:1785 - 1788 vol.3

[\[Abstract\]](#) [\[PDF Full-Text \(372 KB\)\]](#) **IEEE CNF****2 Efficient reordering of Prolog programs***Gooley, M.M.; Wah, B.W.;*

Knowledge and Data Engineering, IEEE Transactions on , Volume: 1 , Issue:

4 , Dec 1989

Pages:470 - 482

[\[Abstract\]](#) [\[PDF Full-Text \(1228 KB\)\]](#) **IEEE JNL****3 Physical storage management for interactive multimedia information systems***Chen, Y.-T.; Kashyap, R.L.; Ghafoor, A.;*

Systems, Man and Cybernetics, 1992., IEEE International Conference on , 18-21

Oct. 1992

Pages:1 - 6 vol.1

[\[Abstract\]](#) [\[PDF Full-Text \(492 KB\)\]](#) **IEEE CNF****4 Adaptive projection operators in multiresolution scientific visualization***Ohlberger, M.; Rumpf, M.;*

Visualization and Computer Graphics, IEEE Transactions on , Volume: 5 , Issue:

1 , Jan.-March 1999

Pages:74 - 94

[\[Abstract\]](#) [\[PDF Full-Text \(2204 KB\)\]](#) [IEEE JNL](#)

5 Adaptive projection operators in multiresolution scientific visualization

Ohlberger, M.; Rumpf, M.;

Visualization and Computer Graphics, IEEE Transactions on , Volume: 4 , Issue:

4 , Oct.-Dec. 1998

Pages:344 - 364

[\[Abstract\]](#) [\[PDF Full-Text \(3432 KB\)\]](#) [IEEE JNL](#)

6 Restricted simple disjunctive decompositions based on grouping symmetric variables

Sawada, H.; Yamashita, S.; Nagoya, A.;

VLSI, 1997. Proceedings. Seventh Great Lakes Symposium on , 13-15 March 1997

Pages:39 - 44

[\[Abstract\]](#) [\[PDF Full-Text \(492 KB\)\]](#) [IEEE CNF](#)

7 Forward model checking techniques oriented to buggy designs

Iwashita, H.; Nakata, T.;

Computer-Aided Design, 1997. Digest of Technical Papers., 1997 IEEE/ACM

International Conference on , 9-13 Nov. 1997

Pages:400 - 404

[\[Abstract\]](#) [\[PDF Full-Text \(420 KB\)\]](#) [IEEE CNF](#)

8 MAFIA: a maximal frequent itemset algorithm for transactional databases

Burdick, D.; Calimlim, M.; Gehrke, J.;

Data Engineering, 2001. Proceedings. 17th International Conference on , 2-6 April 2001

Pages:443 - 452

[\[Abstract\]](#) [\[PDF Full-Text \(740 KB\)\]](#) [IEEE CNF](#)

9 Transistor level synthesis for static CMOS combinational circuits

Liu, C.-P.L.; Abraham, J.A.;

VLSI, 1999. Proceedings. Ninth Great Lakes Symposium on , 4-6 March 1999

Pages:172 - 175

[\[Abstract\]](#) [\[PDF Full-Text \(116 KB\)\]](#) [IEEE CNF](#)

10 Lexical postprocessing by heuristic search and automatic determination of the edit costs

Weigel, A.; Baumann, S.; Rohrschneider, J.;

Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on , Volume: 2 , 14-16 Aug. 1995

Pages:857 - 860 vol.2

[\[Abstract\]](#) [\[PDF Full-Text \(392 KB\)\]](#) [IEEE CNF](#)

11 On variable ordering of binary decision diagrams for the application of multi-level logic synthesis

Fujita, M.; Matsunaga, Y.; Kakuda, T.;
Design Automation. EDAC. Proceedings of the European Conference on , 25-28
Feb. 1991
Pages:50 - 54

[Abstract] [PDF Full-Text (348 KB)] IEEE CNF

12 Variable ordering algorithms for ordered binary decision diagrams and their evaluation

Fujita, M.; Fujisawa, H.; Matsunaga, Y.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions
on , Volume: 12 , Issue: 1 , Jan. 1993
Pages:6 - 12

[Abstract] [PDF Full-Text (588 KB)] IEEE JNL

13 EZW algorithm using depth-first representation of the wavelet zerotree

Li-Minn Ang; Hon Nin Cheung; Eshraghian, K.;
Signal Processing and Its Applications, 1999. ISSPA '99. Proceedings of the Fifth
International Symposium on , Volume: 1 , 22-25 Aug. 1999
Pages:75 - 78 vol.1

[Abstract] [PDF Full-Text (316 KB)] IEEE CNF

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved

THE ACM DIGITAL LIBRARY

Terms used graph manipulation AND depth first

Found 13,204 of 134,837

Sort results
by  [Save results to a Binder](#)Display
results  [Search Tips](#) [Open results in a new window](#) [Try an Advanced Search](#)[Try this search in The ACM Guide](#)

Results 1 - 20 of 200

Result page: 1 [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale **1 Structuring depth-first search algorithms in Haskell**

David J. King, John Launchbury

January 1995 **Proceedings of the 22nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages**Full text available:  [pdf\(1.10 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Depth-first search is the key to a wide variety of graph algorithms. In this paper we express depth-first search in a lazy functional language, obtaining a linear-time implementation. Unlike traditional imperative presentations, we use the structuring methods of functional languages to construct algorithms from individual reusable components. This style of algorithm construction turns out to be quite amenable to formal proof, which we exemplify through a calculational-style proof of a far f ...

2 Symbolic Boolean manipulation with ordered binary-decision diagrams

Randal E. Bryant

September 1992 **ACM Computing Surveys (CSUR)**, Volume 24 Issue 3Full text available:  [pdf\(2.12 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Ordered Binary-Decision Diagrams (OBDDs) represent Boolean functions as directed acyclic graphs. They form a canonical representation, making testing of functional properties such as satisfiability and equivalence straightforward. A number of operations on Boolean functions can be implemented as graph algorithms on OBDD data structures. Using OBDDs, a wide variety of problems can be solved through symbolic analysis. First, the possible variations in system parameters and op ...

Keywords: Boolean algebra, Boolean functions, binary-decision diagrams, branching programs, symbolic analysis, symbolic manipulation

3 Transitive closure algorithms based on graph traversal

Yannis Ioannidis, Raghu Ramakrishnan, Linda Winger

September 1993 **ACM Transactions on Database Systems (TODS)**, Volume 18 Issue 3Full text available:  [pdf\(4.34 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Several graph-based algorithms have been proposed in the literature to compute the transitive closure of a directed graph. We develop two new algorithms (Basic_TC and

Gobal_DFTC) and compare the performance of their implementations in a disk-based environment with a well-known graph-based algorithm proposed by Schmitz. Our algorithms use depth-first search to traverse a graph and a technique called marking to avoid processing some of the arcs in the graph. They compute the ...

Keywords: depth-first search, node reachability, path computations, transitive closure

4 Efficient Planarity Testing

John Hopcroft, Robert Tarjan

October 1974 **Journal of the ACM (JACM)**, Volume 21 Issue 4

Full text available:  [pdf\(1.32 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes an efficient algorithm to determine whether an arbitrary graph G can be embedded in the plane. The algorithm may be viewed as an iterative version of a method originally proposed by Auslander and Parter and correctly formulated by Goldstein. The algorithm used depth-first search and has $O(V)$ time and space bounds, where V is the number of vertices in G . An ALGOL implementation of the al ...

5 Functional programming with graphs

Martin Erwig

August 1997 **ACM SIGPLAN Notices**, Proceedings of the second ACM SIGPLAN international conference on Functional programming, Volume 32 Issue 8

Full text available:  [pdf\(1.40 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Graph algorithms expressed in functional languages often suffer from their inherited imperative, state-based style. In particular, this impedes formal program manipulation. We show how to model persistent graphs in functional languages by graph constructors. This provides a decompositional view of graphs which is very close to that of data types and leads to a "more fictional" formulation of graph algorithms. Graph constructors enable the definition of general fold operations for graphs. We pres ...

6 Breadth-first manipulation of very large binary-decision diagrams

Hiroyuki Ochi, Koichi Yasuoka, Shuzo Yajima

November 1993 **Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design**

Full text available:  [pdf\(823.63 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#)

7 Testing graph connectivity

R. Endre Tarjan

April 1974 **Proceedings of the sixth annual ACM symposium on Theory of computing**

Full text available:  [pdf\(572.84 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

An algorithm proposed by Dinic for finding maximum flows in networks and by Hopcroft and Karp for finding maximum bipartite matchings is applied to graph connectivity problems. It is shown that the algorithm requires $O(V1/2E)$ time to find a maximum set of node-disjoint paths in a graph, and $O(V2/3E)$ time to find a maximum set of edge disjoint paths. These bounds are tight. Thus the node connectivity of a graph may be tested in $O(V5/2)$

Keywords: Connectivity, Flow, Graph, Matching, Maximum flow, Network

Incremental Context-Dependent Analysis for Language-Based Editors

Thomas Reps, Tim Teitelbaum, Alan Demers

July 1983 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 5 Issue 3

Full text available:  [pdf\(1.76 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

9 Parallel breadth-first BDD construction

Bwolen Yang, David R. O'Hallaron

June 1997 **ACM SIGPLAN Notices**, **Proceedings of the sixth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 32 Issue 7

Full text available:  [pdf\(1.24 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

With the increasing complexity of protocol and circuit designs, formal verification has become an important research area and binary decision diagrams (BDDs) have been shown to be a powerful tool in formal verification. This paper presents a parallel algorithm for BDD construction targeted at shared memory multiprocessors and distributed shared memory systems. This algorithm focuses on improving memory access locality through specialized memory managers and partial breadth-first expansion, and o ...

10 High performance BDD package by exploiting memory hierarchy

Jagesh V. Sanghavi, Rajeev K. Ranjan, Robert K. Brayton, Alberto Sangiovanni-Vincentelli

June 1996 **Proceedings of the 33rd annual conference on Design automation conference**

Full text available:  [pdf\(151.05 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

11 Linear expected-time algorithms for connectivity problems (Extended Abstract)

Richard M. Karp, Robert Endre Tarjan

April 1980 **Proceedings of the twelfth annual ACM symposium on Theory of computing**

Full text available:  [pdf\(717.57 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Researchers in recent years have developed many graph algorithms that are fast in the worst case, but little work has been done on graph algorithms that are fast on the average. (Exceptions include the work of Angluin and Valiant [1], Karp [7], and Schnorr [9].) In this paper we analyze the expected running time of four algorithms for solving graph connectivity problems. Our goal is to exhibit algorithms whose expected time is within a constant factor of optimum and to shed light on the pro ...

12 Graph-theoretic methods in database theory

Mihalis Yannakakis

April 1990 **Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**

Full text available:  [pdf\(1.61 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

13 Structure forest and composition factors for small base groups in nearly linear time

Robert Beals, Ákos Seress

July 1992 **Proceedings of the twenty-fourth annual ACM symposium on Theory of computing**

Full text available:  [pdf\(985.52 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A base of a permutation group G is a subset B of the permutation domain such that only the identity of G fixes B pointwise. The permutation representations of important classes of groups, including all finite simple groups other than the alternating groups, admit $O(\log n)$ size bases, where n is the size of the permutation domain. Groups with very small base ...

14 Implicit enumeration of strongly connected components

Aiguo Xie, Peter A. Beerel

November 1999 **Proceedings of the 1999 IEEE/ ACM international conference on Computer-aided design**

Full text available:  [pdf\(191.12 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents a BDD-based implicit algorithm to compute all maximal strongly connected components of directed graphs. The algorithm iteratively applies reachability analysis and sequentially identifies SCCs. Experiments suggest that the algorithm dramatically outperforms the only existing implicit method which must compute the transitive closure of the adjacency-matrix of the graphs.

15 Efficient solution of systems of Boolean equations

Scott Woods, Giorgio Casinovi

January 1997 **Proceedings of the 1996 IEEE/ ACM international conference on Computer-aided design**

Full text available:   [pdf\(79.46 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)
[Publisher Site](#)

This paper describes an algorithm for the efficient solution of large systems of Boolean equations. The algorithm exploits the fact that, in some cases, the composition operation of Boolean functions represented by BDD's can be performed in a very efficient manner. Thus, the algorithm tries to eliminate as many variables and equations as possible through function composition. When the system can no longer be reduced in this way, the elimination process is continued through the use of Shannon dec ...

Keywords: Boolean equations solution, gate-level logic simulation.

16 A fast algorithm for finding dominators in a flowgraph

Thomas Lengauer, Robert Endre Tarjan

January 1979 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
Volume 1 Issue 1

Full text available:  [pdf\(979.86 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A fast algorithm for finding dominators in a flowgraph is presented. The algorithm uses depth-first search and an efficient method of computing functions defined on paths in trees. A simple implementation of the algorithm runs in $O(m \log n)$ time, where m is the number of edges and n is the number of vertices in the problem graph. A more sophisticated implementation runs in $O(m \log \log n)$.

17 Session 8C: Computing strongly connected components in a linear number of symbolic steps

Raffaella Gentilini, Carla Piazza, Alberto Policriti

January 2003 **Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms**

Full text available:  [pdf\(858.17 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We present an algorithm that computes in a linear number of symbolic steps ($O(\log n)$) the strongly connected components (SCCs) of a graph $G = \langle V, E \rangle$ represented by an Ordered Binary Decision Diagram (OBDD). This result matches the

complexity of the (celebrated) Tarjan's algorithm operating on explicit data structures. To date, the best algorithm for the above problem works in $\Theta(\log n)$ symbolic steps ...

18 Integrating pointer variables into one-way constraint models

Brad Vander Zanden, Brad A. Myers, Dario A. Giuse, Pedro Szekely

June 1994 ACM Transactions on Computer-Human Interaction (TOCHI), Volume 1 Issue 2

Full text available:  pdf(3.71 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Pointer variables have long been considered useful for constructing and manipulating data structures in traditional programming languages. This article discusses how pointer variables can be integrated into one-way constraint models and indicates how these constraints can be usefully employed in user interfaces. Pointer variables allow constraints to model a wide array of dynamic application behavior, simplify the implementation of structured objects and demonstrational systems, and improve ...

Keywords: Garnet, constraints, development tools, incremental algorithms

19 High-density reachability analysis

Kavita Ravi, Fabio Somenzi

December 1995 Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design

Full text available:  pdf(220.69 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

 Publisher Site

We address the problem of reachability analysis for large finite state systems. Symbolic techniques have revolutionized reachability analysis but still have limitations in traversing large systems. We present techniques to improve the symbolic breadth-first traversal and compute a lower bound on the reachable states. We identify the problem as one of density during traversal and our techniques seek to improve the same. Our results show a marked improvement on the existing breadth-first traversal ...

20 The system-oriented editor—a tool for managing large software systems

Dick Scheifstrom

December 1987 Proceedings of the 1987 annual ACM SIGAda international conference on Ada

Full text available:  pdf(433.08 KB)

Additional Information: [full citation](#), [references](#), [index terms](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

Terms used **graph manipulation AND depth first traversal**

Found 18,555 of 134,837

Sort results by

 [Save results to a Binder](#)[Try an Advanced Search](#)

Display results

 [Search Tips](#)[Try this search in The ACM Guide](#) [Open results in a new window](#)

Results 21 - 40 of 200

Result page: [previous](#)

1

2

3

4

5

6

7

8

9

10

[next](#)

Best 200 shown

Relevance scale **21 Breadth-first manipulation of very large binary-decision diagrams**

Hiroyuki Ochi, Koichi Yasuoka, Shuzo Yajima

November 1993 **Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design**Full text available:  [pdf\(823.63 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#)**22 Using generational garbage collection to implement cache-conscious data placement**

Trishul M. Chilimbi, James R. Larus

October 1998 **ACM SIGPLAN Notices, Proceedings of the first international symposium on Memory management**, Volume 34 Issue 3Full text available:  [pdf\(1.20 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The cost of accessing main memory is increasing. Machine designers have tried to mitigate the consequences of the processor and memory technology trends underlying this increasing gap with a variety of techniques to reduce or tolerate memory latency. These techniques, unfortunately, are only occasionally successful for pointer-manipulating programs. Recent research has demonstrated the value of a complementary approach, in which pointer-based data structures are reorganized to improve cache loca ...

Keywords: cache-conscious data placement, garbage collection, object-oriented programs, profiling

23 Improved parallel algorithms for the depth-first search and monotone circuit value problems

Peter Varman, Kshitij Doshi

February 1987 **Proceedings of the 15th annual conference on Computer Science**Full text available:  [pdf\(776.50 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Parallel algorithms for ordered depth-first search (ODFS) and the monotone circuit value (MCV) on parallel random access machines (PRAMs) and single bus multiprocessors are presented. While it is known that these problems are log-space complete for P and hence unlikely to have poly-logarithmic time parallel solutions, parallel algorithms that achieve a speedup linear in the number of processors (albeit within a limited range) are of considerable practical interest. In this paper we present ...

24 Principles of runtime support for parallel processors

R. Mirchandaney, J. H. Saltz, R. M. Smith, D. M. Nico, K. Crowley

June 1988 **Proceedings of the 2nd international conference on Supercomputing**

Full text available:  [pdf\(1.32 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

There exists substantial data level parallelism in scientific problems. The PARTY runtime system is an attempt to obtain efficient parallel implementations for scientific computations, particularly those where the data dependencies are manifest only at runtime. This can preclude compiler based detection of certain types of parallelism. The automated system is structured as follows: An appropriate level of granularity is first selected for the computations. A directed acyclic graph represent ...

25 Graphs and trees: Efficiently mining frequent trees in a forest

Mohammed J. Zaki

July 2002 **Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining**

Full text available:  [pdf\(1.26 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Mining frequent trees is very useful in domains like bioinformatics, web mining, mining semistructured data, and so on. We formulate the problem of mining (embedded) subtrees in a forest of rooted, labeled, and ordered trees. We present TREEMINER, a novel algorithm to discover all frequent subtrees in a forest, using a new data structure called scope-list. We contrast TREEMINER with a pattern matching tree mining algorithm (PATTERNMATCHER). W ...

26 Design patterns for the data structures and algorithms course

Bruno R. Preiss

March 1999 **ACM SIGCSE Bulletin , The proceedings of the thirtieth SIGCSE technical symposium on Computer science education**, Volume 31 Issue 1

Full text available:  [pdf\(545.45 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Design patterns have recently emerged as a vehicle for describing and documenting recurring object-oriented designs. More significantly, they offer up a long-awaited framework for teaching good software design. This paper espouses the use of object-oriented *design patterns* in the teaching of the *second course* in computer science, viz., the data structures and algorithms course. To use design patterns effectively, it is necessary to present the various data structures and algo ...

27 Mixin layers: an object-oriented implementation technique for refinements and collaboration-based designs

Yannis Smaragdakis, Don Batory

April 2002 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,

Volume 11 Issue 2

Full text available:  [pdf\(510.43 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A "refinement" is a functionality addition to a software project that can affect multiple dispersed implementation entities (functions, classes, etc.). In this paper, we examine large-scale refinements in terms of a fundamental object-oriented technique called collaboration-based design. We explain how collaborations can be expressed in existing programming languages or can be supported with new language constructs (which we have implemented as extensions to the Java language). We present a spec ...

Keywords: Collaboration-based design, component-based software, product-line

architectures

28 Auxiliary variables for BDD-based representation and manipulation of Boolean functions

Gianpiero Cabodi, Paolo Camurati, Stefano Quer

July 1998 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**,
Volume 3 Issue 3

Full text available:  [pdf\(419.50 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

BDDs are the state-of-the-art technique for representing and manipulating Boolean functions. Their introduction caused a major leap forward in synthesis, verification, and testing. However, they are often unmanageable because of the large amount of nodes. To attack this problem, we insert auxiliary variables that decompose monolithic BDDs in smaller ones. This method works very well for Boolean function representation. As far as combinational circuits are concerned, representing their funct ...

Keywords: binary decision diagrams, finite state machines, functional decompositions, reachability analysis

29 Parallel execution of prolog programs: a survey

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo

July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
Volume 23 Issue 4

Full text available:  [pdf\(1.95 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

30 A overview of modular smalltalk

Allen Wirfs-Brock, Brian Wilkerson

January 1988 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications**, Volume 23 Issue 11

Full text available:  [pdf\(1.23 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper introduces the programming language Modular Smalltalk, a descendant of the Smalltalk-80 programming language. Modular Smalltalk was designed to support teams of software engineers developing production application programs that can run independently of the environment in which they are developed. We first discuss our motivation for designing Modular Smalltalk. Specifically, we examine the properties of Smalltalk-80 that make it inappropriate for our purposes. We then present an o ...

31 Index-driven similarity search in metric spaces

Gisli R. Hjaltason, Hanan Samet

December 2003 **ACM Transactions on Database Systems (TODS)**, Volume 28 Issue 4

Full text available:  [pdf\(650.64 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Similarity search is a very important operation in multimedia databases and other database applications involving complex objects, and involves finding objects in a data set S similar to a query object q , based on some similarity measure. In this article, we focus on methods for similarity search that make the general assumption that similarity is represented with a distance metric d . Existing methods for handling similarity search in this setting typically fall into one of ...

Keywords: Hierarchical metric data structures, distance-based indexing, nearest neighbor queries, range queries, ranking, similarity searching

32 Integrating pointer variables into one-way constraint models

Brad Vander Zanden, Brad A. Myers, Dario A. Giuse, Pedro Szekely

June 1994 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 1 Issue 2

Full text available:  [pdf\(3.71 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Pointer variables have long been considered useful for constructing and manipulating data structures in traditional programming languages. This article discusses how pointer variables can be integrated into one-way constraint models and indicates how these constraints can be usefully employed in user interfaces. Pointer variables allow constraints to model a wide array of dynamic application behavior, simplify the implementation of structured objects and demonstrational systems, and improve ...

Keywords: Garnet, constraints, development tools, incremental algorithms

33 Approximation and decomposition of binary decision diagrams

Kavita Ravi, Kenneth L. McMillan, Thomas R. Shiple, Fabio Somenzi

May 1998 **Proceedings of the 35th annual conference on Design automation conference**

Full text available:  [pdf\(332.04 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Efficient techniques for the manipulation of Binary Decision Diagrams (BDDs) are key to the success of formal verification tools. Recent advances in reachability analysis and model checking algorithms have emphasized the need for efficient algorithms for the approximation and decomposition of BDDs. In this paper we present a new algorithm for approximation and analyze its performance in comparison with existing techniques. We also introduce a new decomposition algorithm that produces balanc ...

34 Elimination algorithms for data flow analysis

Barbara G. Ryder, Marvin C. Paull

September 1986 **ACM Computing Surveys (CSUR)**, Volume 18 Issue 3

Full text available:  [pdf\(3.34 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

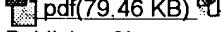
A unified model of a family of data flow algorithms, called elimination methods, is presented. The algorithms, which gather information about the definition and use of data in a program or a set of programs, are characterized by the manner in which they solve the systems of equations that describe data flow problems of interest. The unified model provides implementation-independent descriptions of the algorithms to facilitate comparisons among them and illustrate the sources of improvement ...

35 Efficient solution of systems of Boolean equations

Scott Woods, Giorgio Casinovi

January 1997 **Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design**

Full text available:



Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

[Publisher Site](#)

This paper describes an algorithm for the efficient solution of large systems of Boolean equations. The algorithm exploits the fact that, in some cases, the composition operation of Boolean functions represented by BDD's can be performed in a very efficient manner. Thus, the algorithm tries to eliminate as many variables and equations as possible through function composition. When the system can no longer be reduced in this way, the elimination process is continued through the use of Shannon dec ...

Keywords: Boolean equations solution, gate-level logic simulation.

36 Software infrastructure for parallel visualization: Jupiter: a toolkit for interactive large model visualization

Dirk Bartz, Dirk Steneker, Wolfgang Straßer, Brian Cripe, Tom Gaskins, Kristann Orton, Michael Carter, Andreas Johannsen, Jeff Trom

October 2001 **Proceedings of the IEEE 2001 symposium on parallel and large-data visualization and graphics**

Full text available: [pdf\(1.05 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The fast increasing size of datasets in scientific computing, mechanical engineering, or virtual medicine is quickly exceeding the graphics capabilities of modern computers. Toolkits for the large model visualization address this problem by combining efficient geometric techniques, such as occlusion and visibility culling, mesh reduction, and efficient rendering. In this paper, we introduce *Jupiter*, a toolkit for the interactive visualization of large models which exploits the above mentio ...

Keywords: Large Model Visualization, Toolkit, Visibility, occlusion culling

37 Commutativity analysis: a new analysis framework for parallelizing compilers

Martin C. Rinard, Pedro C. Diniz

May 1996 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1996 conference on Programming language design and implementation**, Volume 31 Issue 5

Full text available: [pdf\(1.79 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents a new analysis technique, commutativity analysis, for automatically parallelizing computations that manipulate dynamic, pointer-based data structures. Commutativity analysis views the computation as composed of operations on objects. It then analyzes the program at this granularity to discover when operations commute (i.e. generate the same final result regardless of the order in which they execute). If all of the operations required to perform a given computation commute, th ...

38 Static grouping of small objects to enhance performance of a paged virtual memory

James W. Stamos

May 1984 **ACM Transactions on Computer Systems (TOCS)**, Volume 2 Issue 2

Full text available: [pdf\(1.79 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

Keywords: Smalltalk, initial placement, object-oriented, paging, programing restructuring, reference trace compression, static grouping, virtual memory

39 **Efficient Planarity Testing**

John Hopcroft, Robert Tarjan

October 1974 **Journal of the ACM (JACM)**, Volume 21 Issue 4

Full text available:  [pdf\(1.32 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes an efficient algorithm to determine whether an arbitrary graph G can be embedded in the plane. The algorithm may be viewed as an iterative version of a method originally proposed by Auslander and Parter and correctly formulated by Goldstein. The algorithm used depth-first search and has $O(V)$ time and space bounds, where V is the number of vertices in G . An ALGOL implementation of the al ...

40 **Provably efficient scheduling for languages with fine-grained parallelism**

Guy E. Blelloch, Phillip B. Gibbons, Yossi Matias

July 1995 **Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures**

Full text available:  [pdf\(1.62 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Results 21 - 40 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)